Manage files and directories

ONTAP 9.8 REST API reference

NetApp
May 08, 2024

Table of Contents

Manage files and directories

Storage volumes volume.uuid files path endpoint overview

Overview

This API is used to read a file, write to a file, retrieve a list of files and directories, and retrieve or modify certain properties of files and directories. The path field is used to specify the path to the directory or file to be acted on. The path field requires using "%2E" to represent "." and "%2F" to represent "/" for the path provided.

File data

Read and write data from/to a named file. To read a file, the Accept request HTTP header must be specified as multipart/form-data, and a value for the `length` query property, which represents the number of bytes to be read, must be specified. The API will fail if the length of data being read/written exceeds 1 MB. This API should only be used on normal files or streams associated with files. The results for other file types, such as LUNs is undefined.

The following APIs are used to read or write data to a file:

– GET /api/storage/volumes/{volume.uuid}/files/{path}?byte_offset=0&length=40 -H "Accept: multipart/form-data"

– POST /api/storage/volumes/{volume.uuid}/files/{path} -H "Content-Type: multipart/form-data" --form "file=the data to be written to the new file"

– PATCH /api/storage/volumes/{volume.uuid}/files/{path}?byte_offset=10 -H "Content-Type: multipart/form-data" --form "file=the new data to be written or overwritten to the existing file starting at byte_offset"

Listing directories and files

A list of files and directories and their properties can be retrieved for a specified path.

The following APIs are used to view a list of files and directories:

– GET /api/storage/volumes/{volume.uuid}/files

– GET /api/storage/volumes/{volume.uuid}/files/{path}

– GET /api/storage/volumes/{volume.uuid}/files/{path}?fields=*

File information

The metadata and detailed information about a single directory or file can be retrieved by setting the `return_metadata` query property to `true`. The information returned includes type, creation_time, modified_time, changed_time, accessed_time, unix_permissions, ownder_id, group_id, size, hard_links_count, inode_number, is_empty, bytes_used, unique_bytes, inode_generation, is_vm_aligned, is_junction, links, and analytics (if requested).

The following API is used to view the properties of a single file or directory:

– GET /api/storage/volumes/{volume.uuid}/files/{path}?return_metadata=true

File usage

Custom details about the usage of a file can be retrieved by specifying a value for the `byte_offset` and `length` query properties.

The following API is used to view the unique bytes, and bytes used, by a file based on the range defined by `byte_offset` and `length`:

– GET /api/storage/volumes/{volume.uuid}/files/{path}?return_metadata=true&byte_offset={int}&length={int}

Create a directory

The following API is used to create a directory:

– POST /api/storage/volumes/{volume.uuid}/files/{path} -d '{ "type" : "directory", "unix-permissions" : "644"}'

Delete an entire directory

A directory can be deleted. The behavior of this call is equivalent to rm -rf.

The following API is used to delete an entire directory:

– DELETE /api/storage/volumes/{volume.uuid}/files/{path}?recurse=true

Delete a file or an empty directory

The following API is used to delete a file or an empty directory:

– DELETE /api/storage/volumes/{volume.uuid}/files/{path}

– DELETE /api/storage/volumes/{volume.uuid}/files/{path}?recurse=false

File system analytics

File system analytics provide a quick method for obtaining information summarizing properties of all files within any directory tree of a volume. When file system analytics are enabled on a volume, `analytics.*` fields may be requested, and will be populated in the response records corresponding to directories. The API does not support file system analytics for requests that are made beyond the boundary of the specified `volume.uuid`.

The following APIs are used to obtain analytics information for a directory:

– GET /api/storage/volumes/{volume.uuid}/files/{path}?fields=analytics

– GET /api/storage/volumes/{volume.uuid}/files/{path}?fields=**

QoS

QoS policies and settings enforce Service Level Objectives (SLO) on a file. A pre-created QoS policy can be used by specifying the `qos.name` or `qos.uuid` properties.

The following APIs are used to assign a QoS policy to a file:

– PATCH /api/storage/volumes/{volume.uuid}/files/{path} -d '{ "qos_policy.name" : "policy" }'

– PATCH /api/storage/volumes/{volume.uuid}/files/{path} -d '{ "qos_policy.uuid" : "b89bc5dd-94a3-11e8-a7a3-0050568edf84" }'

Symlinks

The following APIs are used to create a symlink and read the contents of a symlink:

– POST /api/storage/volumes/{volume.uuid}/files/{path} -d '{ "target" : "directory2/file1" }'

– GET /api/storage/volumes/{volume.uuid}/files/{path}?return_metadata=true&fields=target

Rename a file or a directory

The following API can be used to rename a file or a directory. Note that you need to provide the path relative to the root of the volume in the `path` body parameter.

– PATCH /api/storage/volumes/{volume.uuid}/files/{path} -d '{ "path" : "directory1/directory2" }'

– PATCH /api/storage/volumes/{volume.uuid}/files/{path} -d '{ "path" : "directory1/directory2/file1" }'

Examples

Writing to a new file

```
# The API:
POST /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X POST "https://<mgmt-ip>/api/storage/volumes/54c06ce2-5430-11ea-
90f9-005056a73aff/files/aNewFile" -H "Content-Type: multipart/form-data"
--form "file=the data to be written to the new file"
```

Writing to an existing file

```
# The API:
PATCH /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/storage/volumes/54c06ce2-5430-11ea-
90f9-005056a73aff/files/aNewFile?byte_offset=39" -H "Content-Type:
multipart/form-data" --form "file=*here is a little more data"
```

Reading a file

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/54c06ce2-5430-11ea-
90f9-005056a73aff/files/aNewFile?byte_offset=0&length=100" -H "Accept:
multipart/form-data"

# Response for file data:
--ec51b3541741ade7
Content-Disposition: form-data; name="bytes_read"
Content-Type: text/plain
66
--ec51b3541741ade7
Content-Disposition: form-data; filename="aNewFile"
Content-Type: application/octet-stream
the data to be written to the new file*here is a little more data
--ec51b3541741ade7--
```

Creating a directory

You can use the POST request to create a directory.

```
# The API:
POST /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X POST "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/dir1"  -H 'accept: application/hal+json' -d '{
"type" : "directory", "unix_permissions" : "644" }'

# The response:
{
"num_records": 1,
"records": [
  {
    "path": "dir1",
    "type": "directory",
    "unix_permissions": 644
  }
]
 }
```

Creating a stream on a file

```
# The API:
POST /api/storage/volumes/{volume.uuid}/files/{path}?overwrite=true

# The call:
curl -X POST "https://<mgmt-ip>/api/storage/volumes/54c06ce2-5430-11ea-
90f9-005056a73aff/files/aNewFile?overwrite=true&byte_offset=-
1&stream_name=someStream" -H "Content-Type: multipart/form-data" --form
"file=the data to be written to the new file"
```

Retrieving the list of files in a directory

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/d1%2Fd2%2Fd3"

# Response for file records:
{
"records": [
  {
    "path": "d1/d2/d3",
    "name": ".",
    "type": "directory",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3%2F%2E"
      },
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2F%2E?return_metadata=true"
      }
    }
  },
  {
    "path": "d1/d2/d3",
    "name": "..",
    "type": "directory",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
```

```
005056aca658/files/d1%2Fd2%2Fd3%2F%2E%2E"
      },
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2F%2E%2E?return_metadata=true"
      }
    }
  },
  {
    "path": "d1/d2/d3",
    "name": "f1",
    "type": "file",
    "_links": {
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2File1?return_metadata=true"
      }
    }
  },
  {
    "path": "d1/d2/d3",
    "name": "d5",
    "type": "directory",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3%2Fd5"
      },
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2Fd5?return_metadata=true"
      }
    }
  }
],
"num_records": 4,
"_links": {
  "self": {
    "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3"
  }
}
}
```

Retrieving a list of files based on file type

You can filter the list of files you retrieve based on multiple file types by including a query parameter in the following format type="file\|symlink"

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/d1%2Fd2%2Fd3?type=file&#124;directory"

# Response for file records:
{
"records": [
  {
    "path": "d1/d2/d3",
    "name": ".",
    "type": "directory",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3%2F%2E"
      },
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2F%2E?return_metadata=true"
      }
    }
  },
  {
    "path": "d1/d2/d3",
    "name": "..",
    "type": "directory",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3%2F%2E%2E"
      },
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2F%2E%2E?return_metadata=true"
      }
    }
  },
  {
    "path": "d1/d2/d3",
```

```
    "name": "f1",
    "type": "file",
    "_links": {
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2File1?return_metadata=true"
      }
    }
  },
  {
    "path": "d1/d2/d3",
    "name": "d5",
    "type": "directory",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3%2Fd5"
      },
      "metadata": {
        "href": "/api/storage/volumes/e8274d79-3bba-11ea-b780-
005056a7d72a/files/d1%2Fd2%2Fd3%2Fd5?return_metadata=true"
      }
    }
  }
],
"num_records": 4,
"_links": {
  "self": {
    "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d1%2Fd2%2Fd3"
  }
}
}
```

Retrieving the properties of a directory or a file

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}?return_metadata=true

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/d1%2Fd2%2Fd3%2Ff1?return_metadata=true"

# Response for file properties:
{
"records": [
  {
    "path": "d1/d2/d3/f1",
    "name": "",
    "type": "file",
    "creation_time": "2019-06-12T21:27:28-04:00",
    "modified_time": "2019-06-12T21:27:28-04:00",
    "changed_time": "2019-06-12T21:27:28-04:00",
    "accessed_time": "2019-06-12T21:27:28-04:00",
    "unix_permissions": 644,
    "owner_id": 54738,
    "group_id": 30,
    "size": 200,
    "hard_links_count": 1,
    "inode_number": 1233,
    "bytes_used": 4096,
    "unique_bytes": 4096,
    "inode_generation": 214488325,
    "is_vm_aligned": false,
    "is_junction": false
  }
],
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/storage/volumes/da8bb06c-823e-11e9-b790-
005056acdcb0/files/d1%2Fd2%2Fd3%2Ff1?return_metadata=true"
  }
}
}
```

Creating a symlink to a relative path

You can use the POST request to create a symlink.

```
# The API:
POST /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X POST "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/symlink1"  -H 'accept: application/hal+json' -d '{
"target" : "d1/f1"}'

# The response:
{
"num_records": 1,
"records": [
  {
    "path": "symlink1",
    "target": "d1/f1"
  }
]
}
```

Retrieving the target of a symlink

You can use the GET request to view the target of a symlink.

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/symlink1?return_metadata=true&fields=target"

# The response:
{
"records": [
  {
    "path": "symlink1",
    "target": "d1/f1"
  }
],
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/storage/volumes/54c06ce2-5430-11ea-90f9-
005056a73aff/files/symlink1?return_metadata=true&fields=target"
  }
}
}
```

Retrieving the usage information for a file

You can use the GET request to retrieve the unique bytes held in a file with or without specifing the offset.

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/f1?return_metadata=true&byte_offset=100&length=200"

# The response:
{
"records": [
  {
    "path": "d1/d2/d3/f1",
    "type": "file",
    "creation_time": "2019-06-12T21:27:28-04:00",
    "modified_time": "2019-06-12T21:27:28-04:00",
    "changed_time": "2019-06-12T21:27:28-04:00",
    "accessed_time": "2019-06-12T21:27:28-04:00",
    "unix_permissions": 644,
    "owner_id": 54738,
    "group_id": 30,
    "size": 200,
    "hard_links_count": 1,
    "inode_number": 1233,
    "bytes_used": 4096,
    "unique_bytes": 4096,
    "inode_generation": 214488325,
    "is_vm_aligned": false,
    "is_junction": false
  }
],
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/storage/volumes/cb6b139-8d21-11e9-b926-
05056aca658/files/f1?return_metadata=true&byte_offset=100&length=200"
  }
}
}
```

Retrieving all information (including analytics) for a directory

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}
```

```
# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/1ef5d1b2-f9d7-11e9-
8043-00505682f860/files/d1?return_metadata=true&fields=**"

# Response for all fields of the directory:
{
"records": [
  {
    "svm": {
      "uuid": "58a996a2-f9d5-11e9-8043-00505682f860",
      "_links": {
        "self": {
          "href": "/api/svm/svms/58a996a2-f9d5-11e9-8043-00505682f860"
        }
      }
    },
    "volume": {
        "uuid": "1ef5d1b2-f9d7-11e9-8043-00505682f860",
        "_links": {
          "self": {
            "href": "/api/storage/volumes/1ef5d1b2-f9d7-11e9-8043-
00505682f860"
          }
      }
    },
    "path": "d1",
    "type": "directory",
    "creation_time": "2019-10-28T23:04:13+00:00",
    "modified_time": "2019-10-28T23:10:30+00:00",
    "changed_time": "2019-10-28T23:10:30+00:00",
    "accessed_time": "2019-10-28T23:10:38+00:00",
    "unix_permissions": 755,
    "owner_id": 1002,
    "group_id": 65533,
    "size": 4096,
    "hard_links_count": 5,
    "inode_number": 96,
    "is_empty": false,
    "bytes_used": 4096,
    "inode_generation": 214514951,
    "is_vm_aligned": false,
    "is_junction": false,
    "analytics": {
      "file_count": 668,
      "bytes_used": 209657856,
      "subdir_count": 18,
```

```
      "by_modified_time": {
        "bytes_used": {
          "values": [
            0,
            0,
            0,
            0,
            3112960,
            0,
            14041088,
            20545536,
            0,
            57933824,
            61947904,
            68804608,
            188686336,
            0,
            0,
            0,
            20971520,
            0
          ],
          "percentages": [
            0,
            0,
            0,
            0,
            1.48,
            0,
            6.7,
            9.8,
            0,
            27.63,
            29.55,
            32.82,
            90,
            0,
            0,
            0,
            10,
            0
          ],
          "labels": [
            "2019-W42",
            "2019-W41",
            "2019-W40",
```

```
          "2019-W39",
          "2019-W38",
          "2019-10",
          "2019-09",
          "2019-08",
          "2019-Q4",
          "2019-Q3",
          "2019-Q2",
          "2019-Q1",
          "2019",
          "2018",
          "2017",
          "2016",
          "--2015",
          "unknown"
        ]
      }
    },
    "by_accessed_time": {
      "bytes_used": {
        "values": [
          102760448,
          1867776,
          1245184,
          2179072,
          1556480,
          105873408,
          9027584,
          8093696,
          105873408,
          23969792,
          32382976,
          26460160,
          188686336,
          0,
          0,
          0,
          20971520,
          0
        ],
        "percentages": [
          49.01,
          0.89,
          0.59,
          1.04,
          0.74,
```

```
            50.5,
            4.31,
            3.86,
            50.5,
            11.43,
            15.45,
            12.62,
            90,
            0,
            0,
            0,
            10,
            0
          ],
          "labels": [
            "2019-W42",
            "2019-W41",
            "2019-W40",
            "2019-W39",
            "2019-W38",
            "2019-10",
            "2019-09",
            "2019-08",
            "2019-Q4",
            "2019-Q3",
            "2019-Q2",
            "2019-Q1",
            "2019",
            "2018",
            "2017",
            "2016",
            "--2015",
            "unknown"
          ]
        }
      }
    }
  }
],
"num_records": 1,
"_links": {
  "self": {
    "href": "/api/storage/volumes/1ef5d1b2-f9d7-11e9-8043-
00505682f860/files/d1?return_metadata=true&fields=**"
  }
}
```

```
}
```

Retrieving file system analytics information for a set of histogram buckets

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-
05056aca658/files/d3?type=directory&fields=analytics&analytics.histogram_b
y_time_labels=2019-Q3,2019-Q2,2019-Q1,2018-Q4"

# Response with analytics data
{
"records": [
  {
    "path": "d3",
    "name": ".",
    "type": "directory",
    "analytics": {
      "file_count": 44,
      "bytes_used": 244240384,
      "subdir_count": 14,
      "by_modified_time": {
        "bytes_used": {
          "values": [
            57344,
            29720576,
            196141056,
            57344
          ],
          "percentages": [
            0.02,
            12.17,
            80.31,
            0.02
          ]
        }
      },
      "by_accessed_time": {
        "bytes_used": {
          "values": [
            69632,
            244170752,
```

```
            0,
            0
          ],
          "percentages": [
            0.03,
            99.97,
            0,
            0
          ]
        }
      }
    },
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3%2F%2E"
      },
      "metadata": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3%2F%2E?return_metadata=true"
      }
    }
  },
  {
    "path": "d3",
    "name": "..",
    "type": "directory",
    "analytics": {
      "file_count": 515,
      "bytes_used": 3034574848,
      "subdir_count": 23,
      "by_modified_time": {
        "bytes_used": {
          "values": [
            61440,
            1756479488,
            214622208,
            1191936
          ],
          "percentages": [
            0,
            57.88,
            7.07,
            0.04
          ]
        }
```

```
      },
      "by_accessed_time": {
        "bytes_used": {
          "values": [
            282624,
            3034292224,
            0,
            0
          ],
          "percentages": [
            0.01,
            99.99,
            0,
            0
          ]
        }
      }
    },
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3%2F%2E%2E"
      },
      "metadata": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3%2F%2E%2E?return_metadata=true"
      }
    }
  },
  {
    "path": "d3",
    "name": "d5",
    "type": "directory",
    "analytics": {
      "file_count": 10,
      "bytes_used": 47648768,
      "subdir_count": 4,
      "by_modified_time": {
        "bytes_used": {
          "values": [
            0,
            29638656,
            0,
            0
          ],
          "percentages": [
```

```
            0,
            62.20,
            0,
            0
          ]
        }
      },
      "by_accessed_time": {
        "bytes_used": {
          "values": [
            0,
            47648768,
            0,
            0
          ],
          "percentages": [
            0,
            100,
            0,
            0
          ]
        }
      }
    },
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3%2Fd5"
      },
      "metadata": {
        "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3%2Fd5?return_metadata=true"
      }
    }
  }
],
"num_records": 3,
"analytics": {
  "by_modified_time": {
    "bytes_used": {
      "labels": [
        "2019-Q3",
        "2019-Q2",
        "2019-Q1",
        "2018-Q4"
      ]
```

```
    }
  },
  "by_accessed_time": {
    "bytes_used": {
      "labels": [
        "2019-Q3",
        "2019-Q2",
        "2019-Q1",
        "2018-Q4"
      ]
    }
  }
},
"_links": {
  "self": {
    "href": "/api/storage/volumes/cb6b1b39-8d21-11e9-b926-
005056aca658/files/d3?type=directory&fields=analytics&analytics.histogram_
by_time_labels=2019-Q3,2019-Q2,2019-Q1,2018-Q4"
  }
}
}
```

Identifying the largest subdirectories

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/1ef5d1b2-f9d7-11e9-
8043-
00505682f860/files/d1?fields=analytics.bytes_used&type=directory&order_by=
analytics.bytes_used%20desc"

# Response with the largest subdirectories sorted by their usage:
{
"records": [
  {
    "path": "d1",
    "name": "..",
    "type": "directory",
    "analytics": {
      "bytes_used": 56623104
    }
  },
  {
```

```
    "path": "d1",
    "name": ".",
    "type": "directory",
    "analytics": {
      "bytes_used": 35651584
    }
  },
  {
    "path": "d1",
    "name": "biggest",
    "type": "directory",
    "analytics": {
      "bytes_used": 17825792
    }
  },
  {
    "path": "d1",
    "name": "bigger",
    "type": "directory",
    "analytics": {
      "bytes_used": 10485760
    }
  },
  {
    "path": "d1",
    "name": "big",
    "type": "directory",
    "analytics": {
      "bytes_used": 5242880
    }
  }
],
"num_records": 5,
"_links": {
  "self": {
    "href": "/api/storage/volumes/1ef5d1b2-f9d7-11e9-8043-
00505682f860/files/d1?fields=analytics.bytes_used&type=directory&order_by=
analytics.bytes_used%20desc"
  }
}
}
```


Assigning a QoS policy to a file

You can use the PATCH request to assign a QoS policy to a file.

```
# The API:
PATCH /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X PATCH "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/directory1%2Ffile1" -d '{ "qos_policy": { "name":
"policy" }}'

# The response:
{}
```

Retrieving QoS information for a file

You can use the GET request for all fields with return_metadata="true" to retrieve QoS information for the file.

```
# The API:
GET /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X GET "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/file?return_metadata=true&fields=**"

# The response:
{
"records": [
  {
    "svm": {
     "uuid": "42ee3002-67dd-11ea-8508-005056a7b8ac"
    },
    "volume": {
      "uuid": "c05eb66a-685f-11ea-8508-005056a7b8ac"
    },
    "path": "file",
    "type": "lun",
    "creation_time": "2020-03-17T10:58:40-04:00",
    "modified_time": "2020-03-24T18:15:40-04:00",
    "changed_time": "2020-03-24T18:15:40-04:00",
    "accessed_time": "2020-03-24T18:15:40-04:00",
    "unix_permissions": 644,
    "owner_id": 0,
    "group_id": 0,
    "size": 1048576,
    "hard_links_count": 2,
    "inode_number": 96,
    "bytes_used": 1056768,
    "inode_generation": 219748425,
    "is_vm_aligned": false,
    "is_junction": false,
    "is_snapshot": false,
    "qos_policy": {
      "name": "pg1",
      "uuid": "00725264-688f-11ea-8f10-005056a7b8ac"
    }
  }
] ,
"num_records": 1
}
```

Deleting an entire directory

You can use the DELETE request to remove an entire directory recursively.

```
# The API:
DELETE /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/directory1%2Fdirectory2?recurse=true"

# The response:
{
"job": {
  "uuid": "27d287e8-fcd4-11e9-b8a4-005056a7b97b",
  "_links": {
    "self": {
      "href": "/api/cluster/jobs/27d287e8-fcd4-11e9-b8a4-005056a7b97b"
    }
  }
}
}
```

Deleting an entire directory with specified throttling threshold

You can specify the maximum number of directory delete operations per second when removing an entire directory recursively.

```
# The API:
DELETE /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/directory1%2Fdirectory2?recurse=true&throttle-
deletion=100"

# The response:
{
"job": {
  "uuid": "27d287e8-fcd4-11e9-b8a4-005056a7b97b",
  "_links": {
    "self": {
      "href": "/api/cluster/jobs/27d287e8-fcd4-11e9-b8a4-005056a7b97b"
    }
  }
}
}
```

Deleting an empty directory

You can use the DELETE request to remove an empty directory.

```
# The API:
DELETE /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/directory1%2Fdirectory2"

# The response:
{}
```

Deleting a file

You can use the DELETE request to remove a file.

```
# The API:
DELETE /api/storage/volumes/{volume.uuid}/files/{path}

# The call:
curl -X DELETE "https://<mgmt-ip>/api/storage/volumes/cb6b1b39-8d21-11e9-
b926-05056aca658/files/directory1%2Ffile2"

# The response:
{}
```

Delete an existing file or directory

DELETE `/storage/volumes/{volume.uuid}/files/{path}`

Introduced In: 9.8

Deletes an existing file or directory. Query-based DELETE operations are not supported.

Parameters

Name	Type	In	Required	Description
volume.uuid	string	path	True	Volume UUID
path	string	path	True	The relative path of a directory in the volume. The path field requires using "%2E" to represent "." and "%2F" to represent "/" for the path provided.
recurse	boolean	query	False	Delete an entire directory. The behaviour of this call is equivalent to rm -rf. • Default value:

Name	Type	In	Required	Description
throttle-deletion	integer	query	False	The maximum number of directory delete operations per second. A valid throttle-deletion number is an interger from 10 to 100000.
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202. • Default value: 1 • Max value: 120 • Min value: 0

Response

```
Status: 200, Ok
```

Response

```
Status: 202, Accepted
```

Error

```
Status: Default
```

ONTAP Error Response Codes

Error Code	Description
131074	No such file or directory.
131102	Read-only file system.
131138	Directory not empty.
918235	A volume with UUID {volume.uuid} was not found.
6488081	The {field} field is not supported for DELETE operations.

Name	Type	Description
error	error	

Example error

```
{
  "error": {
    "arguments": {
      "code": "string",
      "message": "string"
    },
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Retrieve files and directories

GET `/storage/volumes/{volume.uuid}/files/{path}`

Introduced In: 9.7

Retrieves a list of files and directories for a given directory or returns only the properties of a single given directory or file of a volume.

Expensive properties

There is an added cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query property. See Requesting specific fields to learn more.

- `analytics`
- `qos_policy.name`
- `qos_policy.uuid`

Parameters

Name	Type	In	Required	Description
volume.uuid	string	path	True	Volume UUID
path	string	path	True	Relative path of a file or directory in the volume. The path field requires using "%2E" to represent "." and "%2F" to represent "/" for the path provided.
byte_offset	integer	query	False	The file offset to start reading from. • Introduced in: 9.8
length	integer	query	False	Length of the range in bytes. • Introduced in: 9.8
return_metadata	boolean	query	False	If true, the request returns metadata for the the directory or file specified in the path. • Introduced in: 9.8 • Default value:
fill_enabled	boolean	query	False	Filter by fill_enabled • Introduced in: 9.8
bytes_used	integer	query	False	Filter by bytes_used
is_snapshot	boolean	query	False	Filter by is_snapshot • Introduced in: 9.8
owner_id	integer	query	False	Filter by owner_id

Name	Type	In	Required	Description
overwrite_enabled	boolean	query	False	Filter by overwrite_enabled • Introduced in: 9.8
unique_bytes	integer	query	False	Filter by unique_bytes • Introduced in: 9.8
changed_time	string	query	False	Filter by changed_time
type	string	query	False	Filter by type
is_junction	boolean	query	False	Filter by is_junction
path	string	query	False	Filter by path
size	integer	query	False	Filter by size
is_vm_aligned	boolean	query	False	Filter by is_vm_aligned
inode_number	integer	query	False	Filter by inode_number
analytics.by_accessed_time.bytes_used.oldest_label	string	query	False	Filter by analytics.by_accessed_time.bytes_used.oldest_label • Introduced in: 9.8
analytics.by_accessed_time.bytes_used.values	integer	query	False	Filter by analytics.by_accessed_time.bytes_used.values • Introduced in: 9.8

Name	Type	In	Required	Description
analytics.by_accessed_time.bytes_used.newest_label	string	query	False	Filter by analytics.by_accessed_time.bytes_used.newest_label • Introduced in: 9.8
analytics.by_accessed_time.bytes_used.labels	string	query	False	Filter by analytics.by_accessed_time.bytes_used.labels • Introduced in: 9.8
analytics.by_accessed_time.bytes_used.percentages	number	query	False	Filter by analytics.by_accessed_time.bytes_used.percentages • Introduced in: 9.8
analytics.file_count	integer	query	False	Filter by analytics.file_count • Introduced in: 9.8
analytics.bytes_used	integer	query	False	Filter by analytics.bytes_used • Introduced in: 9.8
analytics.by_modified_time.bytes_used.oldest_label	string	query	False	Filter by analytics.by_modified_time.bytes_used.oldest_label • Introduced in: 9.8

Name	Type	In	Required	Description
analytics.by_modified_time.bytes_used.values	integer	query	False	Filter by analytics.by_modified_time.bytes_used.values • Introduced in: 9.8
analytics.by_modified_time.bytes_used.newest_label	string	query	False	Filter by analytics.by_modified_time.bytes_used.newest_label • Introduced in: 9.8
analytics.by_modified_time.bytes_used.labels	string	query	False	Filter by analytics.by_modified_time.bytes_used.labels • Introduced in: 9.8
analytics.by_modified_time.bytes_used.percentages	number	query	False	Filter by analytics.by_modified_time.bytes_used.percentages • Introduced in: 9.8
analytics.subdir_count	integer	query	False	Filter by analytics.subdir_count • Introduced in: 9.8
name	string	query	False	Filter by name
target	string	query	False	Filter by target • Introduced in: 9.8
accessed_time	string	query	False	Filter by accessed_time

Name	Type	In	Required	Description
qos_policy.name	string	query	False	Filter by qos_policy.name • Introduced in: 9.8
qos_policy.uuid	string	query	False	Filter by qos_policy.uuid • Introduced in: 9.8
modified_time	string	query	False	Filter by modified_time
inode_generation	integer	query	False	Filter by inode_generation
unix_permissions	integer	query	False	Filter by unix_permissions
volume.uuid	string	query	False	Filter by volume.uuid
volume.name	string	query	False	Filter by volume.name
hard_links_count	integer	query	False	Filter by hard_links_count
group_id	integer	query	False	Filter by group_id
is_empty	boolean	query	False	Filter by is_empty
creation_time	string	query	False	Filter by creation_time

Name	Type	In	Required	Description
analytics.histogram_by_time_labels	array[string]	query	False	Request that returned analytics_histogram_by_time objects including values associated with the specified labels. As described in the object description, the labels may take the following forms: partial-date <tt>--</tt> partial-date partial-date <tt>--</tt> partial-date<tt>-</tt>-partial-date <tt>unknown</tt> Intervals that the system would not normally return may be specified. In this case, the appropriate values and percentages summarizing all files with a time-based attribute within the indicated period of time are calculated and returned in the response. However, there are some restrictions: Any partial-date specified as the beginning or end of an interval must be tracked by the system. Valid partial-dates may be determined by making an OPTIONS request to the <tt>/storage/volumes/{volume.uuid}/files/

Name	Type	In	Required	Description
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. • Default value: 1
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. • Default value: 1 • Max value: 120 • Min value: 0
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

```
Status: 200, Ok
```

Name	Type	Description
_links	_links	

Name	Type	Description
analytics	analytics	Additional file system analytics information that is invariant amongst all elements in the collection. This property is only populated if file system analytics is enabled on the containing volume. This analytics object captures properties that are invariant amongst all elements included in the `records` array. The invariant properties are included here, rather than within the information for each element, to avoid returning an excessive amount of duplicated information when the collection is large.
num_records	integer	Number of records.
records	array[file_info]	

Example response


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "analytics": {
    "by_accessed_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ]
      }
    },
    "by_modified_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ]
      }
    }
  },
  "records": {
    "_links": {
      "metadata": {
        "href": "/api/resourcelink"
      },
      "self": {
        "href": "/api/resourcelink"
```

```
    }
  },
  "accessed_time": "2019-06-12T11:00:16-04:00",
  "analytics": {
    "by_accessed_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "newest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "oldest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "percentages": [
          "0.1",
          "11.24",
          "0.18",
          "15.75",
          "0.75",
          "83.5",
          "0"
        ],
        "values": [
          "15925248",
          "1735569408",
          "27672576",
```

```
            "2430595072",
            "116105216",
            "12889948160",
            "0"
          ]
        }
      },
      "by_modified_time": {
        "bytes_used": {
          "labels": [
            "2019-07",
            "2019-06",
            "2019-05",
            "2019",
            "2018",
            "--2017",
            "unknown"
          ],
          "newest_label": [
            "2019-07",
            "2019-06",
            "2019-05",
            "2019",
            "2018",
            "--2017",
            "unknown"
          ],
          "oldest_label": [
            "2019-07",
            "2019-06",
            "2019-05",
            "2019",
            "2018",
            "--2017",
            "unknown"
          ],
          "percentages": [
            "0.1",
            "11.24",
            "0.18",
            "15.75",
            "0.75",
            "83.5",
            "0"
          ],
          "values": [
```

```
            "15925248",
            "1735569408",
            "27672576",
            "2430595072",
            "116105216",
            "12889948160",
            "0"
          ]
        }
      },
      "bytes_used": "15436648448",
      "file_count": "21134",
      "subdir_count": "35"
    },
    "bytes_used": "4096",
    "changed_time": "2019-06-12T11:00:16-04:00",
    "creation_time": "2019-06-12T11:00:16-04:00",
    "group_id": "30",
    "hard_links_count": "1",
    "inode_generation": "214753547",
    "inode_number": "1695",
    "is_empty": "",
    "is_junction": "",
    "is_snapshot": "",
    "is_vm_aligned": "",
    "modified_time": "2019-06-12T11:00:16-04:00",
    "name": "test_file",
    "owner_id": "54738",
    "path": "d1/d2/d3",
    "qos_policy": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "qos1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "size": "200",
    "target": "some_directory/some_other_directory/some_file",
    "type": "file",
    "unique_bytes": "4096",
    "unix_permissions": "0755",
    "volume": {
      "_links": {
        "self": {
```

```
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  }
}
```


Error

```
Status: Default, Error
```

Name	Type	Description
error	error	

Example error


```
{
  "error": {
    "arguments": {
      "code": "string",
      "message": "string"
    },
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```


Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

bytes_used

Number of bytes used on-disk, broken down by date of last access.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

by_accessed_time

File system analytics information, broken down by date of last access.

Name	Type	Description
bytes_used	bytes_used	Number of bytes used on-disk, broken down by date of last access.

bytes_used

Number of bytes used on-disk, broken down by date of last modification.

includes points in time arbitrarily far in the past. When the end is

with any other time period. This usually occurs when the data was at some point associated with a time in the future.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

by_modified_time

File system analytics information, broken down by date of last modification.

Name	Type	Description
bytes_used	bytes_used	Number of bytes used on-disk, broken down by date of last modification.

includes points in time arbitrarily far in the past. When the end is

with any other time period. This usually occurs when the data was at some point associated with a time in the future.

analytics

Additional file system analytics information that is invariant amongst all elements in the collection.

This property is only populated if file system analytics is enabled on the containing volume.

This analytics object captures properties that are invariant amongst all elements included in the `records` array. The invariant properties are included here, rather than within the information for each element, to avoid returning an excessive amount of duplicated information when the collection is large.

Name	Type	Description
by_accessed_time	by_accessed_time	File system analytics information, broken down by date of last access.
by_modified_time	by_modified_time	File system analytics information, broken down by date of last modification.

_links

Name	Type	Description
metadata	href	
self	href	

bytes_used

Number of bytes used on-disk, broken down by date of last access.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
newest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
oldest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
percentages	array[number]	Percentages for this histogram
values	array[integer]	Values for this histogram

bytes_used

Number of bytes used on-disk, broken down by date of last modification

The "unknown" label tracks data that could not be associated with any other time period. This usually occurs when the data was at some point associated with a time in the future.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
newest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
oldest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
percentages	array[number]	Percentages for this histogram
values	array[integer]	Values for this histogram

analytics

Additional file system analytics information summarizing all descendents of a directory.

This property is only populated if file system analytics is enabled on the containing volume.

In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.

Name	Type	Description
by_accessed_time	by_accessed_time	File system analytics information, broken down by date of last access.
by_modified_time	by_modified_time	File system analytics information, broken down by date of last modification.
bytes_used	integer	Number of bytes used on-disk
file_count	integer	Number of descendants
subdir_count	integer	Number of sub directories

_links

Name	Type	Description
self	href	

qos_policy

The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both `qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none".

Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files.

Note that a QoS policy can be set on a file, or a file's volume, but not on both.

Name	Type	Description
_links	_links	
name	string	The name of the QoS policy. To remove the file from a QoS policy, set this property to an empty string "" or set it to "none" in a PATCH request.
uuid	string	The unique identifier of the QoS policy. Valid in PATCH.

volume

Name	Type	Description
_links	_links	
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • Introduced in: 9.6

file_info

Information about a single file.

Name	Type	Description
_links	_links	
accessed_time	string	Last access time of the file in date-time format.

Name	Type	Description
analytics	analytics	Additional file system analytics information summarizing all descendents of a directory. This property is only populated if file system analytics is enabled on the containing volume. In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.
bytes_used	integer	The actual number of bytes used on disk by this file. If byte_offset and length parameters are specified, this will return the bytes used by the file within the given range.
changed_time	string	Last time data or attributes changed on the file in date-time format.
creation_time	string	Creation time of the file in date-time format.
fill_enabled	boolean	Returns "true" if the space reservation is enabled. The field overwrite_enabled must also be set to the same value as this field.
group_id	integer	The integer ID of the group of the file owner.
hard_links_count	integer	The number of hard links to the file.
inode_generation	integer	Inode generation number.
inode_number	integer	The file inode number.

Name	Type	Description
is_empty	boolean	Specifies whether or not a directory is empty. A directory is considered empty if it only contains entries for "." and "..". This element is present if the file is a directory. In some special error cases, such as when the volume goes offline or when the directory is moved while retrieving this info, this field might not get set.
is_junction	boolean	Returns "true" if the directory is a junction.
is_snapshot	boolean	Returns "true" if the directory is a Snapshot copy.
is_vm_aligned	boolean	Returns true if the file is vm-aligned. A vm-aligned file is a file that is initially padded with zero-filled data so that its actual data starts at an offset other than zero. The amount by which the start offset is adjusted depends on the vm-align setting of the hosting volume.
modified_time	string	Last data modification time of the file in date-time format.
name	string	Name of the file.
overwrite_enabled	boolean	Returns "true" if the space reservation for overwrites is enabled. The field fill_enabled must also be set to the same value as this field.
owner_id	integer	The integer ID of the file owner.
path	string	Path of the file.

Name	Type	Description
qos_policy	qos_policy	The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both `qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none". Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files. Note that a QoS policy can be set on a file, or a file's volume, but not on both.
size	integer	The size of the file, in bytes.
target	string	The relative or absolute path contained in a symlink, in the form <some>/<path>.</path></some>
type	string	Type of the file.
unique_bytes	integer	Number of bytes uniquely held by this file. If byte_offset and length parameters are specified, this will return bytes uniquely held by the file within the given range.

Name	Type	Description
unix_permissions	integer	UNIX permissions to be viewed as an octal number. It consists of 4 digits derived by adding up bits 4 (read), 2 (write), and 1 (execute). The first digit selects the set user ID(4), set group ID (2), and sticky (1) attributes. The second digit selects permissions for the owner of the file; the third selects permissions for other users in the same group; the fourth selects permissions for other users not in the group.
volume	volume	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Write to an existing file with the supplied data

PATCH `/storage/volumes/{volume.uuid}/files/{path}`

Introduced In: 9.8

Writes to an existing file with the supplied data or modifies the size, name, space reservation information, QoS policy, or hole range information of a file. Query-based PATCH operations are not supported.

Parameters

Name	Type	In	Required	Description
volume.uuid	string	path	True	Volume UUID
path	string	path	True	Relative path of a file in the volume. The path field requires using "%2E" to represent "." and "%2F" to represent "/" for the path provided.
byte_offset	integer	query	False	How many bytes into the file to begin writing. Use -1 to append (default).
overwrite	boolean	query	False	If false, and the file exists, the write will fail. Default is false.
stream_name	string	query	False	Name of stream associated with the file to write data to.
data	string	formData	False	Data to write to the file.

Request Body

Name	Type	Description
_links	_links	
accessed_time	string	Last access time of the file in date-time format.

Name	Type	Description
analytics	analytics	Additional file system analytics information summarizing all descendents of a directory. This property is only populated if file system analytics is enabled on the containing volume. In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.
bytes_used	integer	The actual number of bytes used on disk by this file. If byte_offset and length parameters are specified, this will return the bytes used by the file within the given range.
changed_time	string	Last time data or attributes changed on the file in date-time format.
creation_time	string	Creation time of the file in date-time format.
fill_enabled	boolean	Returns "true" if the space reservation is enabled. The field overwrite_enabled must also be set to the same value as this field.
group_id	integer	The integer ID of the group of the file owner.
hard_links_count	integer	The number of hard links to the file.
inode_generation	integer	Inode generation number.
inode_number	integer	The file inode number.

Name	Type	Description
is_empty	boolean	Specifies whether or not a directory is empty. A directory is considered empty if it only contains entries for "." and "..". This element is present if the file is a directory. In some special error cases, such as when the volume goes offline or when the directory is moved while retrieving this info, this field might not get set.
is_junction	boolean	Returns "true" if the directory is a junction.
is_snapshot	boolean	Returns "true" if the directory is a Snapshot copy.
is_vm_aligned	boolean	Returns true if the file is vm-aligned. A vm-aligned file is a file that is initially padded with zero-filled data so that its actual data starts at an offset other than zero. The amount by which the start offset is adjusted depends on the vm-align setting of the hosting volume.
modified_time	string	Last data modification time of the file in date-time format.
name	string	Name of the file.
overwrite_enabled	boolean	Returns "true" if the space reservation for overwrites is enabled. The field fill_enabled must also be set to the same value as this field.
owner_id	integer	The integer ID of the file owner.
path	string	Path of the file.

Name	Type	Description
qos_policy	qos_policy	The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both `qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none". Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files. Note that a QoS policy can be set on a file, or a file's volume, but not on both.
size	integer	The size of the file, in bytes.
target	string	The relative or absolute path contained in a symlink, in the form <some>/<path>.</path></some>
type	string	Type of the file.
unique_bytes	integer	Number of bytes uniquely held by this file. If byte_offset and length parameters are specified, this will return bytes uniquely held by the file within the given range.

Name	Type	Description
unix_permissions	integer	UNIX permissions to be viewed as an octal number. It consists of 4 digits derived by adding up bits 4 (read), 2 (write), and 1 (execute). The first digit selects the set user ID(4), set group ID (2), and sticky (1) attributes. The second digit selects permissions for the owner of the file; the third selects permissions for other users in the same group; the fourth selects permissions for other users not in the group.
volume	volume	

Example request


```
{
  "_links": {
    "metadata": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "accessed_time": "2019-06-12T11:00:16-04:00",
  "analytics": {
    "by_accessed_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "newest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "oldest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "percentages": [
          "0.1",
          "11.24",
          "0.18",
```

```
          "15.75",
          "0.75",
          "83.5",
          "0"
        ],
        "values": [
          "15925248",
          "1735569408",
          "27672576",
          "2430595072",
          "116105216",
          "12889948160",
          "0"
        ]
      }
    },
    "by_modified_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "newest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "oldest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "percentages": [
```

```
          "0.1",
          "11.24",
          "0.18",
          "15.75",
          "0.75",
          "83.5",
          "0"
        ],
        "values": [
          "15925248",
          "1735569408",
          "27672576",
          "2430595072",
          "116105216",
          "12889948160",
          "0"
        ]
      }
    },
    "bytes_used": "15436648448",
    "file_count": "21134",
    "subdir_count": "35"
  },
  "bytes_used": "4096",
  "changed_time": "2019-06-12T11:00:16-04:00",
  "creation_time": "2019-06-12T11:00:16-04:00",
  "group_id": "30",
  "hard_links_count": "1",
  "inode_generation": "214753547",
  "inode_number": "1695",
  "is_empty": "",
  "is_junction": "",
  "is_snapshot": "",
  "is_vm_aligned": "",
  "modified_time": "2019-06-12T11:00:16-04:00",
  "name": "test_file",
  "owner_id": "54738",
  "path": "d1/d2/d3",
  "qos_policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "qos1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
```

```
  },
  "size": "200",
  "target": "some_directory/some_other_directory/some_file",
  "type": "file",
  "unique_bytes": "4096",
  "unix_permissions": "0755",
  "volume": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  }
}
```


Response

```
Status: 200, Ok
```

Error

```
Status: Default
```

ONTAP Error Response Codes

Error Code	Description
918235	A volume with UUID {volume.uuid} was not found.
6488081	The {field} field is not supported for PATCH operations.
6488082	Failed to rename {path}.
6488083	Failed to rename {path} to {path} because a directory named {path} already exists.

Name	Type	Description
error	error	

Example error


```
{
  "error": {
    "arguments": {
      "code": "string",
      "message": "string"
    },
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```


Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
metadata	href	
self	href	

bytes_used

Number of bytes used on-disk, broken down by date of last access.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
newest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
oldest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
percentages	array[number]	Percentages for this histogram
values	array[integer]	Values for this histogram

by_accessed_time

File system analytics information, broken down by date of last access.

The "unknown" label tracks data that could not be associated with any other time period. This usually occurs when the data was

Name	Type	Description
bytes_used	bytes_used	Number of bytes used on-disk, broken down by date of last access.

bytes_used

Number of bytes used on-disk, broken down by date of last modification.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
newest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
oldest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
percentages	array[number]	Percentages for this histogram
values	array[integer]	Values for this histogram

by_modified_time

File system analytics information, broken down by date of last modification.

The "unknown" label tracks data that could not be associated with any other time period. This usually occurs when the data was

Name	Type	Description
bytes_used	bytes_used	Number of bytes used on-disk, broken down by date of last modification.

analytics

Additional file system analytics information summarizing all descendents of a directory.

This property is only populated if file system analytics is enabled on the containing volume.

In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.

Name	Type	Description
by_accessed_time	by_accessed_time	File system analytics information, broken down by date of last access.
by_modified_time	by_modified_time	File system analytics information, broken down by date of last modification.
bytes_used	integer	Number of bytes used on-disk
file_count	integer	Number of descendants
subdir_count	integer	Number of sub directories

_links

Name	Type	Description
self	href	

qos_policy

The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both

`qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none".

Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files.

Note that a QoS policy can be set on a file, or a file's volume, but not on both.

Name	Type	Description
_links	_links	
name	string	The name of the QoS policy. To remove the file from a QoS policy, set this property to an empty string "" or set it to "none" in a PATCH request.
uuid	string	The unique identifier of the QoS policy. Valid in PATCH.

volume

Name	Type	Description
_links	_links	
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • Introduced in: 9.6

file_info

Information about a single file.

Name	Type	Description
_links	_links	
accessed_time	string	Last access time of the file in date-time format.

Name	Type	Description
analytics	analytics	Additional file system analytics information summarizing all descendents of a directory. This property is only populated if file system analytics is enabled on the containing volume. In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.
bytes_used	integer	The actual number of bytes used on disk by this file. If byte_offset and length parameters are specified, this will return the bytes used by the file within the given range.
changed_time	string	Last time data or attributes changed on the file in date-time format.
creation_time	string	Creation time of the file in date-time format.
fill_enabled	boolean	Returns "true" if the space reservation is enabled. The field overwrite_enabled must also be set to the same value as this field.
group_id	integer	The integer ID of the group of the file owner.
hard_links_count	integer	The number of hard links to the file.
inode_generation	integer	Inode generation number.
inode_number	integer	The file inode number.

Name	Type	Description
is_empty	boolean	Specifies whether or not a directory is empty. A directory is considered empty if it only contains entries for "." and "..". This element is present if the file is a directory. In some special error cases, such as when the volume goes offline or when the directory is moved while retrieving this info, this field might not get set.
is_junction	boolean	Returns "true" if the directory is a junction.
is_snapshot	boolean	Returns "true" if the directory is a Snapshot copy.
is_vm_aligned	boolean	Returns true if the file is vm-aligned. A vm-aligned file is a file that is initially padded with zero-filled data so that its actual data starts at an offset other than zero. The amount by which the start offset is adjusted depends on the vm-align setting of the hosting volume.
modified_time	string	Last data modification time of the file in date-time format.
name	string	Name of the file.
overwrite_enabled	boolean	Returns "true" if the space reservation for overwrites is enabled. The field fill_enabled must also be set to the same value as this field.
owner_id	integer	The integer ID of the file owner.
path	string	Path of the file.

Name	Type	Description
qos_policy	qos_policy	The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both `qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none". Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files. Note that a QoS policy can be set on a file, or a file's volume, but not on both.
size	integer	The size of the file, in bytes.
target	string	The relative or absolute path contained in a symlink, in the form <some>/<path>.</path></some>
type	string	Type of the file.
unique_bytes	integer	Number of bytes uniquely held by this file. If byte_offset and length parameters are specified, this will return bytes uniquely held by the file within the given range.

Name	Type	Description
unix_permissions	integer	UNIX permissions to be viewed as an octal number. It consists of 4 digits derived by adding up bits 4 (read), 2 (write), and 1 (execute). The first digit selects the set user ID(4), set group ID (2), and sticky (1) attributes. The second digit selects permissions for the owner of the file; the third selects permissions for other users in the same group; the fourth selects permissions for other users not in the group.
volume	volume	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Create a new file with the supplied data

POST `/storage/volumes/{volume.uuid}/files/{path}`

Introduced In: 9.8

Creates a new file with the supplied data, creates a new directory or creates a new symlink.

Parameters

Name	Type	In	Required	Description
volume.uuid	string	path	True	Volume UUID
path	string	path	True	Relative path of a new file, directory or symlink. The path field requires using "%2E" to represent "." and "%2F" to represent "/" for the path provided.
byte_offset	integer	query	False	How many bytes into the file to begin writing. Use -1 to append (default).
overwrite	boolean	query	False	If false, and the file exists, the write will fail. Default is false.
stream_name	string	query	False	Name of stream associated with the file to write data to.
data	string	formData	False	Data to write to the file.

Request Body

Name	Type	Description
_links	_links	
accessed_time	string	Last access time of the file in date-time format.

Name	Type	Description
analytics	analytics	Additional file system analytics information summarizing all descendents of a directory. This property is only populated if file system analytics is enabled on the containing volume. In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.
bytes_used	integer	The actual number of bytes used on disk by this file. If byte_offset and length parameters are specified, this will return the bytes used by the file within the given range.
changed_time	string	Last time data or attributes changed on the file in date-time format.
creation_time	string	Creation time of the file in date-time format.
fill_enabled	boolean	Returns "true" if the space reservation is enabled. The field overwrite_enabled must also be set to the same value as this field.
group_id	integer	The integer ID of the group of the file owner.
hard_links_count	integer	The number of hard links to the file.
inode_generation	integer	Inode generation number.
inode_number	integer	The file inode number.

Name	Type	Description
is_empty	boolean	Specifies whether or not a directory is empty. A directory is considered empty if it only contains entries for "." and "..". This element is present if the file is a directory. In some special error cases, such as when the volume goes offline or when the directory is moved while retrieving this info, this field might not get set.
is_junction	boolean	Returns "true" if the directory is a junction.
is_snapshot	boolean	Returns "true" if the directory is a Snapshot copy.
is_vm_aligned	boolean	Returns true if the file is vm-aligned. A vm-aligned file is a file that is initially padded with zero-filled data so that its actual data starts at an offset other than zero. The amount by which the start offset is adjusted depends on the vm-align setting of the hosting volume.
modified_time	string	Last data modification time of the file in date-time format.
name	string	Name of the file.
overwrite_enabled	boolean	Returns "true" if the space reservation for overwrites is enabled. The field fill_enabled must also be set to the same value as this field.
owner_id	integer	The integer ID of the file owner.
path	string	Path of the file.

Name	Type	Description
qos_policy	qos_policy	The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both `qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none". Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files. Note that a QoS policy can be set on a file, or a file's volume, but not on both.
size	integer	The size of the file, in bytes.
target	string	The relative or absolute path contained in a symlink, in the form <some>/<path>.</path></some>
type	string	Type of the file.
unique_bytes	integer	Number of bytes uniquely held by this file. If byte_offset and length parameters are specified, this will return bytes uniquely held by the file within the given range.

Name	Type	Description
unix_permissions	integer	UNIX permissions to be viewed as an octal number. It consists of 4 digits derived by adding up bits 4 (read), 2 (write), and 1 (execute). The first digit selects the set user ID(4), set group ID (2), and sticky (1) attributes. The second digit selects permissions for the owner of the file; the third selects permissions for other users in the same group; the fourth selects permissions for other users not in the group.
volume	volume	

Example request


```
{
  "_links": {
    "metadata": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "accessed_time": "2019-06-12T11:00:16-04:00",
  "analytics": {
    "by_accessed_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "newest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "oldest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "percentages": [
          "0.1",
          "11.24",
          "0.18",
```

```
          "15.75",
          "0.75",
          "83.5",
          "0"
        ],
        "values": [
          "15925248",
          "1735569408",
          "27672576",
          "2430595072",
          "116105216",
          "12889948160",
          "0"
        ]
      }
    },
    "by_modified_time": {
      "bytes_used": {
        "labels": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "newest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "oldest_label": [
          "2019-07",
          "2019-06",
          "2019-05",
          "2019",
          "2018",
          "--2017",
          "unknown"
        ],
        "percentages": [
```

```
          "0.1",
          "11.24",
          "0.18",
          "15.75",
          "0.75",
          "83.5",
          "0"
        ],
        "values": [
          "15925248",
          "1735569408",
          "27672576",
          "2430595072",
          "116105216",
          "12889948160",
          "0"
        ]
      }
    },
    "bytes_used": "15436648448",
    "file_count": "21134",
    "subdir_count": "35"
  },
  "bytes_used": "4096",
  "changed_time": "2019-06-12T11:00:16-04:00",
  "creation_time": "2019-06-12T11:00:16-04:00",
  "group_id": "30",
  "hard_links_count": "1",
  "inode_generation": "214753547",
  "inode_number": "1695",
  "is_empty": "",
  "is_junction": "",
  "is_snapshot": "",
  "is_vm_aligned": "",
  "modified_time": "2019-06-12T11:00:16-04:00",
  "name": "test_file",
  "owner_id": "54738",
  "path": "d1/d2/d3",
  "qos_policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "qos1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
```

```
  },
  "size": "200",
  "target": "some_directory/some_other_directory/some_file",
  "type": "file",
  "unique_bytes": "4096",
  "unix_permissions": "0755",
  "volume": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  }
}
```


Response

```
Status: 201, Created
```

Error

```
Status: Default
```

ONTAP Error Response Codes

Error Code	Description
917505	The SVM does not exist.
917525	The volume in the symlink path does not exist in the SVM.
917698	The volume in the symlink path is not mounted in the namespace.
6488064	This command is not supported.
6488065	The volume in the symlink path is invalid.
6488066	Mounting the unjunctioned volume in the symlink path failed.
6488069	Internal file error.
6488084	Failed to create {path} because the "unix_permissions" field was not specified.

Error Code	Description
6488085	Failed to create {path} because the "type" field was not specified.
8257536	This operation is not supported for the system volume specified in the symlink path.
8257541	Failed to compute the SVM identification from this content.
8257542	This operation is not supported for the administrative SVM.
9437549	This operation is not allowed on SVMs with Infinite Volume.
13172837	This operation is not permitted because the SVM is locked for a migrate operation.

Name	Type	Description
error	error	

Example error

```
{
  "error": {
    "arguments": {
      "code": "string",
      "message": "string"
    },
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
metadata	href	
self	href	

bytes_used

Number of bytes used on-disk, broken down by date of last access.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
newest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
oldest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
percentages	array[number]	Percentages for this histogram
values	array[integer]	Values for this histogram

by_accessed_time

File system analytics information, broken down by date of last access.

The "unknown" label tracks data that could not be associated with any other time period. This usually occurs when the data was

Name	Type	Description
bytes_used	bytes_used	Number of bytes used on-disk, broken down by date of last access.

bytes_used

Number of bytes used on-disk, broken down by date of last modification.

Name	Type	Description
labels	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
newest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
oldest_label	array[string]	Labels for this histogram. Each label is a string indicating the period of time the corresponding data is associated with. Elements of the array take one of the following forms: a partial date in an extended ISO8601 representation an interval between partial dates in an extended ISO8601 representation, where "--" is used to separate the beginning and end of the interval the string literal "unknown" For partial dates and partial date intervals where components of a date are unspecified, the label allows for any valid normalized values the unspecified components might take. For example, the label "2017" allows for any time within the year 2017. Essentially, this is the fully specified interval 2017-01-01T00:00:00—2017-12-31T23:59:59. Similarly, the interval "2018-05—2018-07" allows for any time within the months of May, June, and July in 2018, corresponding to the fully specified interval 2018-05-01T00:00:00—2018-07-31T23:59:59. The following extensions to ISO8601 are used: Quarters may be specified. The form *yyyy*-Q*q* is used to represent the *q*th quarter of the year *yyyy*. Q1 consists of the months January, February, and March; Q2 consists of April, May, and June; Q3 consists of July, August, and September; Q4 consists of October, November, and December. For example, the label "2019-Q2" represents the second quarter of the year 2019, which corresponds to the interval 2019-04-01T00:00:00—2019-06-30T23:59:59. Either the beginning or end of an interval may be omitted. When the beginning is omitted, the interval

Name	Type	Description
percentages	array[number]	Percentages for this histogram
values	array[integer]	Values for this histogram

by_modified_time

File system analytics information, broken down by date of last modification.

The "unknown" label tracks data that could not be associated with any other time period. This usually occurs when the data was

Name	Type	Description
bytes_used	bytes_used	Number of bytes used on-disk, broken down by date of last modification.

analytics

Additional file system analytics information summarizing all descendents of a directory.

This property is only populated if file system analytics is enabled on the containing volume.

In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.

Name	Type	Description
by_accessed_time	by_accessed_time	File system analytics information, broken down by date of last access.
by_modified_time	by_modified_time	File system analytics information, broken down by date of last modification.
bytes_used	integer	Number of bytes used on-disk
file_count	integer	Number of descendants
subdir_count	integer	Number of sub directories

_links

Name	Type	Description
self	href	

qos_policy

The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both

`qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none".

Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files.

Note that a QoS policy can be set on a file, or a file's volume, but not on both.

Name	Type	Description
_links	_links	
name	string	The name of the QoS policy. To remove the file from a QoS policy, set this property to an empty string "" or set it to "none" in a PATCH request.
uuid	string	The unique identifier of the QoS policy. Valid in PATCH.

volume

Name	Type	Description
_links	_links	
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • Introduced in: 9.6

file_info

Information about a single file.

Name	Type	Description
_links	_links	
accessed_time	string	Last access time of the file in date-time format.

Name	Type	Description
analytics	analytics	Additional file system analytics information summarizing all descendents of a directory. This property is only populated if file system analytics is enabled on the containing volume. In the context of the `records` property of a GET /storage/volumes/{volume.uuid}/files/{path} call returns a large collection.
bytes_used	integer	The actual number of bytes used on disk by this file. If byte_offset and length parameters are specified, this will return the bytes used by the file within the given range.
changed_time	string	Last time data or attributes changed on the file in date-time format.
creation_time	string	Creation time of the file in date-time format.
fill_enabled	boolean	Returns "true" if the space reservation is enabled. The field overwrite_enabled must also be set to the same value as this field.
group_id	integer	The integer ID of the group of the file owner.
hard_links_count	integer	The number of hard links to the file.
inode_generation	integer	Inode generation number.
inode_number	integer	The file inode number.

Name	Type	Description
is_empty	boolean	Specifies whether or not a directory is empty. A directory is considered empty if it only contains entries for "." and "..". This element is present if the file is a directory. In some special error cases, such as when the volume goes offline or when the directory is moved while retrieving this info, this field might not get set.
is_junction	boolean	Returns "true" if the directory is a junction.
is_snapshot	boolean	Returns "true" if the directory is a Snapshot copy.
is_vm_aligned	boolean	Returns true if the file is vm-aligned. A vm-aligned file is a file that is initially padded with zero-filled data so that its actual data starts at an offset other than zero. The amount by which the start offset is adjusted depends on the vm-align setting of the hosting volume.
modified_time	string	Last data modification time of the file in date-time format.
name	string	Name of the file.
overwrite_enabled	boolean	Returns "true" if the space reservation for overwrites is enabled. The field fill_enabled must also be set to the same value as this field.
owner_id	integer	The integer ID of the file owner.
path	string	Path of the file.

Name	Type	Description
qos_policy	qos_policy	The QoS policy for the file. Both traditional and adaptive QoS policies are supported. If both `qos_policy.uuid` and `qos_policy.name` properties are specified in the same request, they must refer to the same QoS policy. To remove the file from a QoS policy, set the property `qos_policy.name` in a PATCH request to an empty string "" or "none". Files which are in use as a LUN cannot be assigned to a QoS policy, instead use PATCH on /storage/luns to assign a QoS policy for such files. Note that a QoS policy can be set on a file, or a file's volume, but not on both.
size	integer	The size of the file, in bytes.
target	string	The relative or absolute path contained in a symlink, in the form <some>/<path>.</path></some>
type	string	Type of the file.
unique_bytes	integer	Number of bytes uniquely held by this file. If byte_offset and length parameters are specified, this will return bytes uniquely held by the file within the given range.

Name	Type	Description
unix_permissions	integer	UNIX permissions to be viewed as an octal number. It consists of 4 digits derived by adding up bits 4 (read), 2 (write), and 1 (execute). The first digit selects the set user ID(4), set group ID (2), and sticky (1) attributes. The second digit selects permissions for the owner of the file; the third selects permissions for other users in the same group; the fourth selects permissions for other users not in the group.
volume	volume	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Retrieve historical performance metrics for a volume

GET `/storage/volumes/{volume.uuid}/metrics`

Introduced In: 9.7

Retrieves historical performance metrics for a volume.

Parameters

Name	Type	In	Required	Description
latency.total	integer	query	False	Filter by latency.total
latency.read	integer	query	False	Filter by latency.read
latency.other	integer	query	False	Filter by latency.other
latency.write	integer	query	False	Filter by latency.write
cloud.duration	string	query	False	Filter by cloud.duration
cloud.timestamp	string	query	False	Filter by cloud.timestamp
cloud.latency.total	integer	query	False	Filter by cloud.latency.total
cloud.latency.read	integer	query	False	Filter by cloud.latency.read
cloud.latency.other	integer	query	False	Filter by cloud.latency.other
cloud.latency.write	integer	query	False	Filter by cloud.latency.write
cloud.iops.total	integer	query	False	Filter by cloud.iops.total
cloud.iops.read	integer	query	False	Filter by cloud.iops.read
cloud.iops.other	integer	query	False	Filter by cloud.iops.other
cloud.iops.write	integer	query	False	Filter by cloud.iops.write
cloud.status	string	query	False	Filter by cloud.status

Name	Type	In	Required	Description
flexcache.timestamp	string	query	False	Filter by flexcache.timestamp • Introduced in: 9.8
flexcache.cache_miss_percent	integer	query	False	Filter by flexcache.cache_miss_percent • Introduced in: 9.8
flexcache.duration	string	query	False	Filter by flexcache.duration • Introduced in: 9.8
flexcache.status	string	query	False	Filter by flexcache.status • Introduced in: 9.8
status	string	query	False	Filter by status
timestamp	string	query	False	Filter by timestamp
duration	string	query	False	Filter by duration
throughput.total	integer	query	False	Filter by throughput.total
throughput.read	integer	query	False	Filter by throughput.read
throughput.other	integer	query	False	Filter by throughput.other
throughput.write	integer	query	False	Filter by throughput.write
iops.total	integer	query	False	Filter by iops.total
iops.read	integer	query	False	Filter by iops.read

Name	Type	In	Required	Description
iops.other	integer	query	False	Filter by iops.other
iops.write	integer	query	False	Filter by iops.write
volume.uuid	string	path	True	Unique identifier of the volume.
interval	string	query	False	The time range for the data. Examples can be 1h, 1d, 1m, 1w, 1y. The period for each time range is as follows: • 1h: Metrics over the most recent hour sampled over 15 seconds. • 1d: Metrics over the most recent day sampled over 5 minutes. • 1w: Metrics over the most recent week sampled over 30 minutes. • 1m: Metrics over the most recent month sampled over 2 hours. • 1y: Metrics over the most recent year sampled over a day. • Default value: 1 • enum: ["1h", "1d", "1w", "1m", "1y"]

Name	Type	In	Required	Description
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. • Default value: 1 • Max value: 120 • Min value: 0
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
order_by	array[string]	query	False	Order results by specified fields and optional [asc
desc] direction. Default direction is 'asc' for ascending.	return_records	boolean	query	False

Response

```
Status: 200, Ok
```

Name	Type	Description
_links	_links	
num_records	integer	Number of records
records	array[records]	

Example response


```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "duration": "PT15S",
    "iops": {
      "read": "200",
      "total": "1000",
      "write": "100"
    },
    "latency": {
      "read": "200",
      "total": "1000",
      "write": "100"
    },
    "status": "ok",
    "throughput": {
      "read": "200",
      "total": "1000",
      "write": "100"
    },
    "timestamp": "2017-01-25T11:20:13Z"
  }
}
```


Error

```
Status: Default, Error
```

Name	Type	Description
error	error	

Example error


```
{
  "error": {
    "arguments": {
      "code": "string",
      "message": "string"
    },
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```


Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

iops

The rate of I/O operations observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Peformance metric for write I/O operations.

latency

The round trip latency in microseconds observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Peformance metric for write I/O operations.

throughput

The rate of throughput bytes per second observed at the storage object.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Peformance metric for write I/O operations.

records

Performance numbers, such as IOPS latency and throughput.

Name	Type	Description
_links	_links	

Name	Type	Description
duration	string	The duration over which this sample is calculated. The time durations are represented in the ISO-8601 standard format. Samples can be calculated over the following durations:
iops	iops	The rate of I/O operations observed at the storage object.
latency	latency	The round trip latency in microseconds observed at the storage object.
status	string	Errors associated with the sample. For example, if the aggregation of data over multiple nodes fails, then any partial errors might return "ok" on success or "error" on an internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled to the previous 15 second timestamp and tagged with "backfilled_data". "Inconsistent_ delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "Negative_delta" is returned when an expected monotonically increasing value has decreased in value. "Inconsistent_old_data" is returned when one or more nodes do not have the latest data.
throughput	throughput	The rate of throughput bytes per second observed at the storage object.
timestamp	string	The timestamp of the performance data.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Copyright information

Trademark information